

Claims:

What is claimed is:

1 1. A method for conducting the exchange of data with a terminal-based application
2 program comprising the steps of:

3 (a) mapping a plurality of available states within a terminal data stream of the
4 terminal-based application program to respective discrete state definitions within a finite
5 state machine, including:

6 (1) interpreting any element, terminal command, data item, or sequence of
7 terminal commands and data items within the terminal data stream as a discrete
8 state having a respective one of the state definitions, and

9 (2) using an object model containing a set of interfaces, said interfaces being
10 utilized as the basis for the state definitions; and

11 (b) defining a plurality of state transition rules which are utilized to manipulate the
12 state definitions within the finite state machine.

1 2. The method of claim 1, wherein the object model contains the set of interfaces
2 and a set of classes.

1 3. The method of claim 1, wherein manipulation of the state transition rules allows
2 for two-way communication between a client and the terminal-based application
3 program.

1 4. The method of claim 1, further comprising creating one or more data sets, each
2 comprising a definable set of data elements from within the terminal-based application
3 program.

1 5. The method of claim 1, wherein the object model is a distributed object
2 transaction model, which allows for access to data from the terminal-based application
3 program by any local or remote client service, system, or application.

1 6. The method of claim 1 further comprising unifying and resolving multiple
2 terminal-based applications through unification and resolution of a plurality of instances
3 of a program that performs steps (a) and (b).

1 7. The method of claim 6, further comprising creating a plurality of data sets, each
2 data set being formed through a unification of multiple data sets from the plurality of
3 instances of the program that performs steps (a) and (b).

1 8. The method of claim 6, wherein each object model is a distributed object
2 transaction model which allows for access to data from the terminal-based application
3 program by any local or remote client service, system, or application.

1 9. The method of claim 6, further comprising using an expert system which accesses
2 and resolves data items from the plurality of instances of the program that performs steps
3 (a) and (b) and translates them into cohesive super sets of data.

1 10. The method of claim 9, further comprising using the object model for the creation
2 of data resolution and translation rules.

1 11. The method of claim 9 further comprising using the object model for definition of
2 actions to proactively resolve data errors or discrepancies across the underlying plurality
3 of instances of the program that performs steps (a) and (b).

1 12. The method of claim 1, further comprising altering an interface presented to a
2 user of the terminal-based application program through addition of one of the group
3 consisting of new screens and new data fields within existing screens, wherein the new
4 screens and new data fields are populated with data retrieved from an alternate data
5 source.

1 13. The method of claim 12, further comprising monitoring the terminal data stream.

1 14. The method of claim 13, further comprising:

2 recognizing pre-defined states within the terminal data stream, which define one
3 or more states during a user's interaction with the terminal-based application program;

4 and

5 presenting the new data screens or fields to the user.

1 15. The method of claim 12, wherein the object model describes interaction between
2 the alternate data source and a program that performs steps (a) and (b).

1 16. The method of claim 12, wherein the object model describes:

2 the addition of new user screens or the addition of new data fields to existing
3 application screens, and

4 interaction between the user and the new screens or fields.

1 17. The method of claim 1 further comprising using software tools to automate
2 creation and maintenance of an integration system based on knowledge of a domain of
3 the terminal-based application program.

1 18. A system for conducting the exchange of data with a terminal-based application
2 program comprising:

3 a finite state machine, in which a plurality of available states within a terminal
4 data stream of the terminal-based application program are mapped to respective discrete
5 state definitions, including:

6 means for interpreting any element, terminal command, data item, or
7 sequence of terminal commands and data items within the terminal data stream as
8 a discrete state having a respective one of the state definitions;

9 means for using an object model containing a set of interfaces, said
10 interfaces being utilized as the basis for the state definitions; and

11 means for defining a plurality of state transition rules which are utilized to
12 manipulate the state definitions within the finite state machine.

1 19. The system of claim 18, wherein the object model contains the set of interfaces
2 and a set of classes.

1 20. The system of claim 18, wherein manipulation of the state transition rules allows
2 for two-way communication between a client and the terminal-based application
3 program.

1 21. The system of claim 18, further comprising one or more data sets, each
2 comprising a definable set of data elements from within the terminal-based application
3 program.

1 22. The system of claim 18, wherein the object model is a distributed object
2 transaction model, which allows for access to data from the terminal-based application
3 program by any local or remote client service, system, or application.

1 23. The system of claim 18 wherein multiple terminal-based applications are unified
2 and resolved through unification and resolution of a plurality of instances of the finite
3 state machine.

1 24. The system of claim 23, further comprising a plurality of data sets, each data set
2 being formed through a unification of multiple data sets from the plurality of instances of
3 the finite state machine.

1 25. The system of claim 23, wherein each object model is a distributed object
2 transaction model which allows for access to data from the terminal-based application
3 program by any local or remote client service, system, or application.

1 26. The system of claim 23, further comprising an expert system which accesses and
2 resolves data items from the plurality of instances of the finite state machine and
3 translates them into cohesive super sets of data.

1 27. The system of claim 26, wherein the object model is used for the creation of data
2 resolution and translation rules.

1 28. The system of claim 26 wherein the object model is used for definition of actions
2 to proactively resolve data errors or discrepancies across the underlying plurality of
3 instances of the finite state machine.

1 29. The system of claim 18, further comprising an interface presented to a user of the
2 terminal-based application program, the interface formed through addition of one of the
3 group consisting of new screens and new data fields within existing screens, wherein the
4 new screens and new data fields are populated with data retrieved from an alternate data
5 source.

1 30. The system of claim 29, further comprising means for monitoring the terminal
2 data stream.

1 31. The system of claim 30, wherein:
2 pre-defined states are recognized within the terminal data stream, which define
3 one or more states during a user's interaction with the terminal-based application
4 program; and
5 the new data screens or fields are presented to the user.

1 32. The system of claim 29, wherein the object model describes interaction between
2 the alternate data source and the finite state machine.

1 33. The system of claim 29, wherein the object model describes:
2 the addition of new user screens or the addition of new data fields to existing
3 application screens, and
4 interaction between the user and the new screens or fields.

1 34. The system of claim 18 further comprising software tools that automate creation

2 and maintenance of an integration system based on knowledge of a domain of the
3 terminal-based application program.

1 35. The system of claim 18, further comprising a tool which automates the capture of
2 a terminal data stream and the creation of state definitions for a particular terminal-based
3 application.

1 36. The system of claim 35, wherein the tool allows the user to define the data sets
2 which will be made available.

1 37. The system of claim 36, wherein the tool allows the user to define state transition
2 rules to access and manipulate the data sets, to read and write data elements, using a
3 point-and-click flowchart-style interface.

1 38. The system of claim 35, further comprising software tools which automate
2 creation and maintenance of a unification and resolution system based on a knowledge of
3 underlying integration systems being unified.

1 39. The system of claim 35, wherein the tool allows the user to define the data super-
2 sets which will be made available by the system.

1 40. The system of claim 35, wherein the tool allows the user to create and define data
2 unification and resolution rules.

1 41. The system of claim 35, wherein the tool allows allows the user to define rules to
2 manage data errors and discrepancies in the terminal data stream.